

5

Application for United States Letters Patent


for

**METHOD AND APPARATUS FOR RESTRICTED EXECUTION OF
SECURITY SENSITIVE INSTRUCTIONS**

by

**Brian C. Barnes
Rodney Schmidt
Geoffrey Strongin**

10
15
20
25
30

CERTIFICATE OF MAILING UNDER 37 C.F.R. § 1.10	
EXPRESS MAIL NO.:	EL 522 495 837 US
DATE OF DEPOSIT:	December 3, 2001
<p>I hereby certify that this paper or fee is being deposited with the United States Postal Service "EXPRESS MAIL POST OFFICE TO POST OFFICE" service addressed to: USPTO, Box Patent Application P.O. Box 2327, Arlington, VA 22202, under 35 U.S.C.21(a) and 37 CFR 1.6(e) in response to the Postal Service Interruption designated by the USPTO on November 16, 2001, in place of "EXPRESS MAIL POST OFFICE TO ADDRESSEE" under 37 CFR 1.10 normally addressed to: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.</p> <p> Signature</p>	

METHOD AND APPARATUS FOR RESTRICTED EXECUTION OF SECURITY SENSITIVE INSTRUCTIONS

BACKGROUND OF THE INVENTION

5

1. FIELD OF THE INVENTION

This invention relates generally to computers, and, more particularly, to a method and apparatus for ensuring the secure operation of a computer system.

10

2. DESCRIPTION OF THE RELATED ART

15
20
25

General purpose computing systems, such as personal computers, have evolved from performing a single task to performing a multitude of tasks executed simultaneously. Systems that multitask require security and protection services to protect their operating system from user processes, and to protect the processes from each other. Without protection, a rogue program, for example, could unintentionally destroy the program code or data in the memory space belonging to the operating system or to another process.

Typically, in x86 microprocessor environments, security features have been implemented to reduce the possibility of a rogue program by providing varying privilege levels. Different types of software run at these varying privilege levels, and, thus, have varying access to the resources of the computing system. As illustrated in Figure 1, the operating system of the computer typically runs at the highest privilege level (i.e., level 0), which generally permits the operating system free access to virtually any of the system resources of the computer system. Software drivers are also permitted to run at the highest privilege level. Application programs, on the other hand, typically run at the lowest privilege

level (i.e., level 3), and are generally permitted access to the system resources, but only by the permission of the operating system. Thus, at least theoretically, the operating system prevents one application program from accessing the system resources assigned to another application program. For example, the operating system assigns space in physical memory to the application programs, which are generally free to read and write within the assigned space only. That is, the operating system normally prevents one application program from modifying the contents of a portion of the physical memory that has been assigned to another application program.

Security defects and bugs, however, have inevitably penetrated their way into the operating system and drivers. Accordingly, it is possible that rogue programs or drivers could accidentally or intentionally use these security defects and bugs to gain access to portions of the memory from which they were intended to be excluded. Once a program or driver has access to otherwise protected memory, it can impede or even take over operation of the targeted software, including any peripheral devices associated therewith. For example, the operation of a modem in a data processor could be subverted by a rogue program that gains access to the memory space of the application program or driver originally associated with the modem. The rogue program could then instruct the modem to place unauthorized and undesirable calls over a telephone network, or to otherwise interfere with the proper operation of the telephone network.

The most recent version of Microsoft's Windows® operating system, Windows 2000®, now has over one million lines of code contained in its kernel and associated kernel-mode drivers. Thus, more than one million lines of code have generally free access to the system resources. Accordingly, owing to the sheer size of the operating system, additional

defects and bugs will inevitably be uncovered. Thus, rogue programs may very well continue to exploit yet-to-be discovered security defects and bugs to accidentally or intentionally gain access to portions of memory from which they were intended to be shielded.

- 5 The present invention is directed to overcoming, or at least reducing the effects of, one or more of the problems set forth above.

1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808 2809 2810 2811 2812 2813 2814 2815 2816 2817 2818 2819 2820 2821 2822 2823 2824 2825 2826 2827 2828 2829 2830 2831 2832 2833 2834 2835 2836 2837 2838 2839 2840 2841 2842 2843 2844 2845 2846 2847 2848 2849 2850 2851 2852 2853 2854 2855 2856 2857 2858 2859 2860 2861 2862 2863 2864 2865 2866 2867 2868 2869 2870 2871 2872 2873 2874 2875 2876 2877 2878 2879 2880 2881 2882 2883 2884 2885 2886 2887 2888 2889 2890 2891 2892 2893 2894 2895 2896 2897 2898 2899 2900 2901 2902 2903 2904 2905 2906 2907 2908 2909 2910 2911 2912 2913 2914 2915 2916 2917 2918 2919 2920 2921 2922 2923 2924 2925 2926 2927 2928 2929 2930 2931 2932 2933 2934 2935 2936 2937 2938 2939 2940 2941 2942 2943 2944 2945 2946 2947 2948 2949 2950 2951 2952 2953 2954 2955 2956 2957 2958 2959 2960 2961 2962 2963 2964 2965 2966 2967 2968 2969 2970 2971 2972 2973 2974 2975 2976 2977 2978 2979 2980 2981 2982 2983 2984 2985 2986 2987 2988 2989 2990 2991 2992 2993 2994 2995 2996 2997 2998 2999 3000 3001 3002 3003 3004 3005 3006 3007 3008 3009 3010 3011 3012 3013 3014 3015 3016 3017 3018 3019 3020 3021 3022 3023 3024 3025 3026 3027 3028 3029 3030 3031 3032 3033 3034 3035 3036 3037 3038 3039 3040 3041 3042 3043 3044 3045 3046 3047 3048 3049 3050 3051 3052 3053 3054 3055 3056 3057 3058 3059 3060 3061 3062 3063 3064 3065 3066 3067 3068 3069 3070 3071 3072 3073 3074 3075 3076 3077 3078 3079 3080 3081 3082 3083 3084 3085 3086 3087 3088 3089 3090 3091 3092 3093 3094 3095 3096 3097 3098 3099 3100 3101 3102 3103 3104 3105 3106 3107 3108 3109 3110 3111 3112 3113 3114 3115 3116 3117 3118 3119 3120 3121 3122 3123 3124 3125 3126 3127 3128 3129 3130 3131 3132 3133 3134 3135 3136 3137 3138 3139 3140 3141 3142 3143 3144 3145 3146 3147 3148 3149 3150 3151 3152 3153 3154 3155 3156 3157 3158 3159 3160 3161 3162 3163 3164 3165 3166 3167 3168 3169 3170 3171 3172 3173 3174 3175 3176 3177 3178 3179 3180 3181 3182 3183 3184 3185 3186 3187 3188 3189 3190 3191 3192 3193 3194 3195 3196 3197 3198 3199 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3

SUMMARY OF THE INVENTION

In one aspect of the present invention, a method is provided. The method includes associating a first security identification (ID) with each of a plurality of instructions or a set of instructions that are to be executed by a processor. A request is made to execute at least one of the plurality of instructions or set of instructions by a software code running on the processor. A second security ID associated with the software code running on the processor is obtained. A comparison is made between the second security ID with the first security ID. The requested instruction or set of instructions is executed providing that the second security ID matches the first security ID.

In another aspect of the invention, an apparatus is provided. The apparatus comprises a processor for running code thereon, and for associating a first security identification (ID) with each of a plurality of instructions or a set of instructions that are to be executed by the processor. The processor receives a request to execute at least one of the plurality of instructions or set of instructions by the code running thereon, obtains a second security ID associated with the code, compares the second security ID with the first security ID, and executes the requested instruction or set of instructions providing that the second security ID matches the first security ID.

In another aspect of the invention, an article is provided. The article comprises one or more machine-readable storage media including instructions that when executed enable a processor to associate a first security identification (ID) with each of a plurality of instructions or a set of instructions that are to be executed by the processor, and request to execute at least one of the plurality of instructions or set of instructions by software code

running on the processor. A second security ID associated with the software code running on the processor is obtained. A comparison is made between the second security ID with the first security ID. The requested instruction or set of instructions is executed providing that the second security ID matches the first security ID.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be understood by reference to the following description taken in
5 conjunction with the accompanying drawings, in which the leftmost significant digit(s) in the
reference numerals denote(s) the first figure in which the respective reference numerals
appear, and in which:

Figure 1 schematically illustrates one embodiment of a conventional security scheme
10 implemented in a computer system;

Figure 2 shows a simplified block diagram of a computer system including a
processor unit that employs a security access scheme according to one embodiment of the
present invention;

Figure 3 provides a more detailed representation of the processor unit provided in
15 Figure 2;

Figure 4 illustrates a more detailed representation of a system memory controller that
is provided in Figure 3; and

Figures 5A and 5B illustrate a process for restricting the execution of security
sensitive instructions according to one embodiment of the present invention.

20 While the invention is susceptible to various modifications and alternative forms,
specific embodiments thereof have been shown by way of example in the drawings and are
herein described in detail. It should be understood, however, that the description herein of
specific embodiments is not intended to limit the invention to the particular forms disclosed,

but, on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

5 Illustrative embodiments of the invention are described below. In the interest of clarity, not all features of an actual implementation are described in this specification. It will of course be appreciated that in the development of any such actual embodiment, numerous implementation-specific decisions must be made to achieve the developers' specific goals, such as compliance with system-related and business-related constraints, which will vary from one implementation to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking for those of ordinary skill in the art having the benefit of this disclosure.

Turning now to the drawings, and specifically referring to Figure 2, a simplified block diagram of a computer system 200 for restricting the execution of security sensitive instructions is shown according to one embodiment of the present invention. In the illustrated embodiment, the computer system 200 in one of its simplest forms comprises a processor unit 205, at least one input device 210 and at least one output device 215. The input device 210 may include, for example, a keyboard, mouse, touchpad, touchscreen and the like. The output device 215 may include a monitor, printer, and/or various other devices to convey information to the system 200's user that is processed by the processor unit 205.

In accordance with the illustrated embodiment, the processor unit 205 employs a security access scheme, which restricts the system 200 from executing certain types of security sensitive instructions by the processor unit 205. The security sensitive instructions

may include those instructions or set of instructions where it is desired to limit their execution by certain types of code running on the processor unit 205.

An input by the system 200's user via the input device 210 may initiate the execution of one or more software structures or codes in the processor unit 205. The memory resident in the processor unit 205 is then accessed to execute the various software structures or codes to implement the desired task. The security access scheme employed by the processor unit 205 may restrict the execution of some security sensitive instructions that are to be carried out by the processor 205 based upon predetermined security entries programmed therein, and, accordingly, prohibit certain computing tasks from being executed by the processor unit 205.

Turning now to Figure 3, a more detailed representation of the processor unit 205 of the system 200 is provided. In accordance with one embodiment of the present invention, the processor unit 205 comprises a processor 305, which takes the form of a microprocessor in the illustrated embodiment. It will be appreciated, however, that the processor 305 may take the form of various other processor types and may also include multiple processors without departing from the spirit and scope of the present invention. The processor unit 205 further includes a system memory controller 310 and a memory module 315. According to the illustrated embodiment, the memory module 315 includes a physical memory 350, which may take the form of a synchronous dynamic random access memory (SDRAM) or double-rate dynamic random access memory (DDRAM). It will be appreciated, however, that the physical memory 350 may take the form of a variety of other memory types, and, thus, need not be limited to the aforementioned examples.

The processor 305 communicates with the memory module 315 via the system memory controller 310 through a host bus 330. According to one embodiment, the system

memory controller 310 provides memory addresses and logic signals to the memory module 315 to characterize the desired memory transactions by the processor 305.

A primary bus 335 that is used to communicate with peripheral devices 345 is also coupled to the host bus 330 through a host bridge 340. In one embodiment, the primary bus 335 takes the form of a peripheral component interconnect (PCI) bus (see PCI Specification, Rev. 2.1). It will also be appreciated that the processor unit 205 may further be configured to include other buses, such as a secondary PCI bus (not shown) or various other peripheral devices (not shown), without departing from the spirit and scope of the present invention.

In one embodiment, the processor 305 performs a plurality of computer processing operations based upon executable instructions. In response to a request to run a particular instruction by software code that is running thereon, the processor 305 performs a memory access procedure to carry out the appropriate execution prompted by the code. The processor 305 performs memory access via the system memory controller 310, which provides access to the physical memory 350 of the memory module 315 to perform the desired task upon execution of the particular instruction.

Turning now to Figure 4, a more detailed representation of the system memory controller 310 is provided according to one embodiment of the present invention. The system memory controller 310 comprises a virtual memory access table 405 and a virtual memory 410. The processor 305 uses the virtual memory access table 405 to derive a virtual (or linear) memory address, which indicates a particular location within the virtual memory 410. Once the particular virtual memory address is determined, the virtual memory 410 is accessed to determine a physical memory address corresponding to the virtual memory address obtained from the virtual memory access table 405. The physical memory address obtained

from the virtual memory 410 points to a particular memory location within the physical memory 350, which provides the memory space needed to perform the desired instruction by the software code that is currently running on the processor 305. It will be appreciated that the process for obtaining the virtual memory address (as described above) is provided in its most simplistic form and that various other tables may be accessed in addition to the virtual memory access table 405 to derive the virtual memory address. These particular procedures for deriving the virtual memory address are known to those skilled in the art. Accordingly, the specific details of such procedures are not provided herein to avoid unnecessarily obscuring the present invention.

According to one embodiment, a security identification (ID) may be associated with a particular security sensitive instruction or set of instructions that are executable by the processor 305. In one embodiment, the security ID may be a fixed value, or alternatively, may be stored in a programmable register (not shown) that may be resident in the processor 305. According to the illustrated embodiment, the security sensitive instructions may include moves to CR3 that define the virtual memory base, moves to CR0 that define the operating mode of the processor 305, and/or moves to CR4 that define various sub-modes of the processor 305. Additionally, any instructions that are configured to modify a programmable register defining the security ID for the various executable instructions may also be considered as security sensitive. It will be appreciated that the aforementioned list of examples is not necessarily exhaustive. Accordingly, various other types of instructions performed by the processor 305 may be considered security sensitive, and, thus, need not be limited to the examples provided above.

The security ID that is associated with the software code (e.g., driver, service, kernel routine, application code, etc.) currently being executed by the processor 305 is

obtained by referencing the address of the virtual memory table 405 that corresponds to the security sensitive instruction or set of instructions that the code desires to execute. When a particular security sensitive instruction is requested by the currently executed code on the processor 305, the security ID of the currently executing code obtained by referencing the address of the virtual memory access table 405 is compared by the processor 305 to the security ID for the particular security instruction or set of instructions that the code desires to execute. As mentioned, the security ID of the security sensitive instruction(s) may be a fixed value or may be alterable and stored in a programmable register. If a match of the security IDs exist, the processor 305 executes the security sensitive instruction or set of instructions as requested by the software code running thereon. If, on the other hand, the security ID of the currently executing code and the security ID of the desired security sensitive instruction(s) do not match, the processor 305 denies the execution of the security sensitive instruction or set of instructions as requested by the software code running on the processor 305.

Turning now to Figure 5, a process 500 for restricting the execution of security sensitive instructions by code running on the processor 305 is provided according to one embodiment. The process 500 commences at block 505 where an instruction or a set (i.e., grouping) of instructions are classified as security sensitive, and each such instruction or set of instructions is associated with a security ID at the processor 305. In accordance with one embodiment, the security ID may be a constant value (i.e., the security sensitive instructions may have a fixed security ID) associated with the security sensitive instruction(s). In an alternative embodiment, the security ID of the security sensitive instruction may be stored in a programmable register (not shown) at the processor 305, which the operating system running thereon may modify. It will be appreciated that other methods for associating a security ID with a security sensitive instruction(s) may be performed, in addition to the

examples provided above, without departing from the spirit and scope of the present invention.

At block 510, the processor 305 determines whether the software code currently running thereon is attempting to execute an instruction that has been classified as security sensitive. In one embodiment, the processor 305 may determine if the requested instruction is security sensitive by referencing the programmable register of the processor 305, for example, to see if a security ID is associated therewith. If the software code currently running on the processor 305 does not attempt to execute a security sensitive instruction (i.e., no security ID is associated with the requested instruction) at block 510, the processor 305 executes the instruction without invoking a security procedure at block 515 (i.e., the instruction is executed in a usual manner by the processor 305). If, however, the software code currently running on the processor 305 attempts to execute a security sensitive instruction in block 510, a security procedure is invoked by the processor 305 at block 520 to determine whether the code running on the processor 305 is authorized to execute the security sensitive instruction.

The process 500 continues at block 525 where a security ID of the software code running on the processor 305 is obtained by referencing an address of the virtual memory table 405 corresponding to the security sensitive instruction(s) that the code desires to execute. After obtaining the security ID associated with the software code running thereon, the processor 305 obtains the security ID of the security sensitive instruction(s) that the code desires to execute at block 530. In one embodiment, the security ID of the security sensitive instruction(s) may be obtained from a programmable register (not shown).

Subsequent to retrieving the security ID associated with the particular instruction from the virtual memory table 405, the processor 305 compares the security ID associated

with the code running on the processor 305 and the security ID associated with the particular instruction at block 535. At block 540, the processor 305 determines whether the security ID associated with the code running thereon matches the security ID associated with the particular instruction that the code is attempting to execute.

5 If there is a mismatch between the security ID associated with the code running on the processor and the security ID associated with the particular instruction, the processor 305 denies execution of the security sensitive instruction by the software code running thereon at block 545. If there is a match between the security ID associated with the code running on the processor 305 and the security ID associated with the security sensitive instruction, the processor 305 executes the security sensitive instruction at block 550. According to one embodiment, when the security IDs match, the virtual memory address corresponding to the software code to be executed by the processor 305 is obtained from the virtual memory access table 405, the physical address is obtained from the virtual memory 410 using the virtual memory address, and the memory space within the physical memory 350 is made accessible to the processor 305 to execute the desired security sensitive instruction.

In accordance with one embodiment, a precise match between the security IDs is needed to execute the security sensitive instruction by the processor 305. It will be appreciated, however, that alternatively the security IDs may be broken up into fields, thus providing a multi-layered security scheme. Accordingly, although the software codes and security sensitive instructions may have unique security IDs in their entirety, a common field or component within the security ID would permit executing the desired security sensitive instruction. Thus, the security ID may be configured to establish and indicate classes, families or other grouping information such that a particular class of codes may be able to

execute a particular instruction or set of instructions because a portion of their security IDs match.

For example, in one embodiment, the security ID may comprise a class and subclass field, and if the class field of the security IDs match, the subclass field of the security IDs may be disregarded (ignored) for comparison purposes between the security IDs. Accordingly, in this particular embodiment, execution of the security sensitive instruction(s) by the processor 305 may be permitted providing that at least the class fields of the security IDs match. In another embodiment, the comparison of the security IDs (or a portion thereof, such as a class field, for example) need not necessarily be an exact arithmetic match. For example, if the class field of the security ID associated with the code running on the processor 305 is greater than (or, alternatively, less than) or equal to the class code of the security ID associated with the security sensitive instruction(s), then execution of the security sensitive instruction may be permitted. It will be appreciated, however, that the particular format to the security IDs need not necessarily be limited to the examples provided above.

The particular embodiments disclosed above are illustrative only, as the invention may be modified and practiced in different but equivalent manners apparent to those skilled in the art having the benefit of the teachings herein. Furthermore, no limitations are intended to the details of construction or design herein shown, other than as described in the claims below. It is therefore evident that the particular embodiments disclosed above may be altered or modified and all such variations are considered within the scope and spirit of the invention. Accordingly, the protection sought herein is as set forth in the claims below.